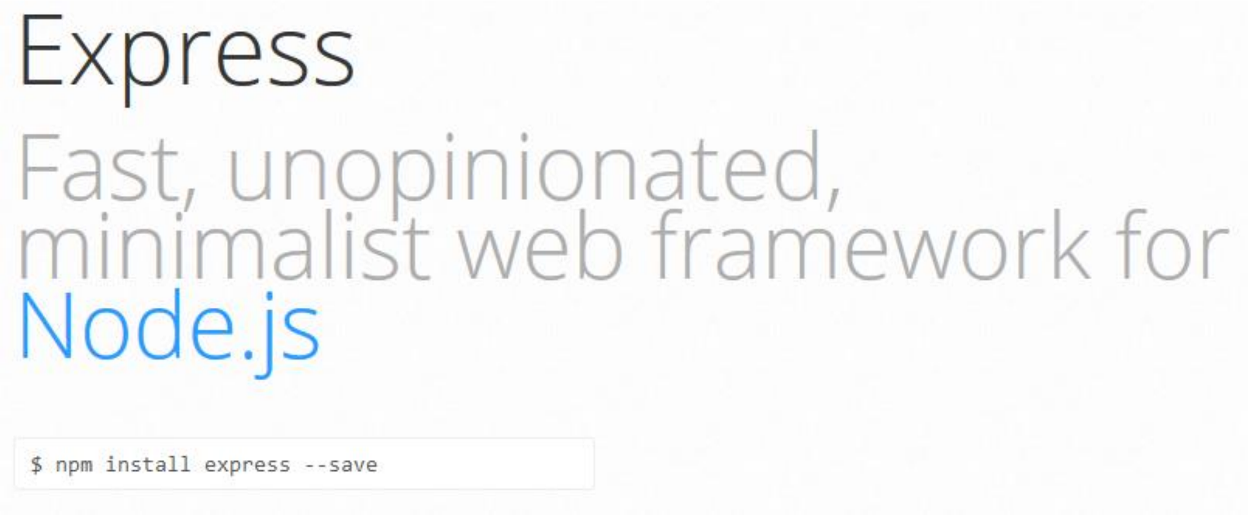
EXPRESS – WEB FRAMEWORK FOR NODEJS

Express là một Web Framework chạy trên nền NodeJS, hỗ trợ chúng ta rất mạnh về Routing, Middleware, quản lý Request & Response rất tốt và còn rất nhiều chức năng khác.

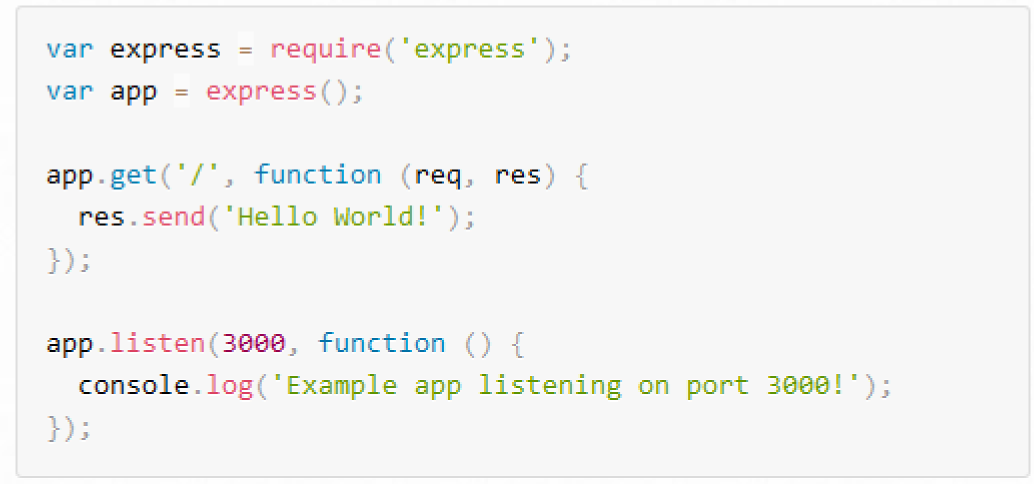


Express

Fast, unopinionated, minimalist web framework for Node.js

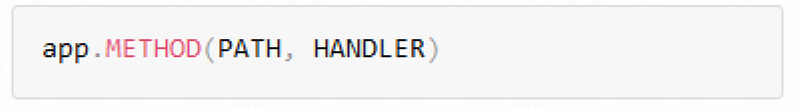
```
$ npm install express --save
```

1 – Cài đặt & cấu hình Express

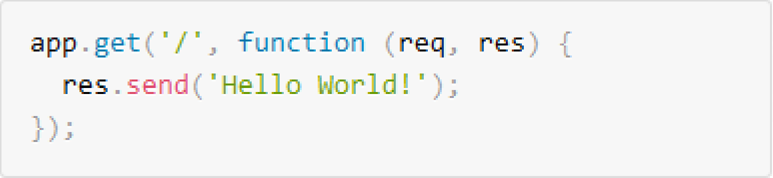


```
var express = require('express');
var app = express();

app.get('/', function (req, res) {
  res.send('Hello World!');
});

app.listen(3000, function () {
  console.log('Example app listening on port 3000!');
});
```

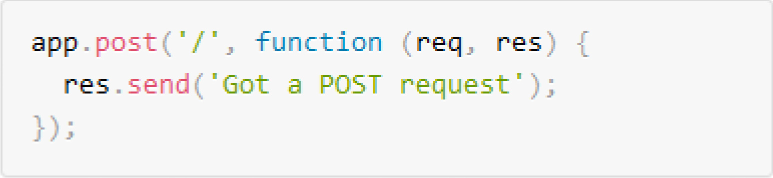
2 - Basic route



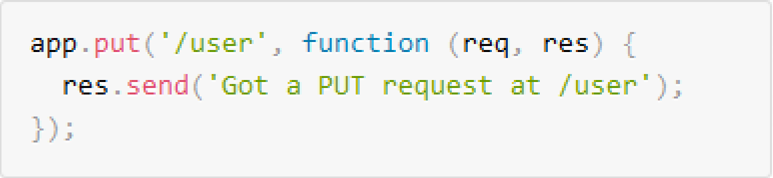
```
app.METHOD(PATH, HANDLER)
```



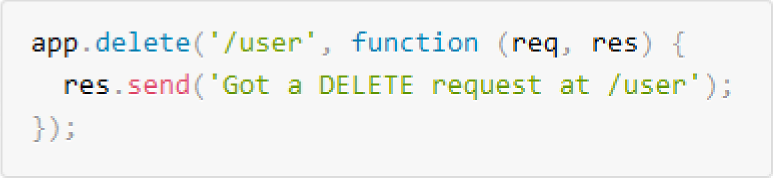
```
app.get('/', function (req, res) {
  res.send('Hello World!');
});
```



```
app.post('/', function (req, res) {
  res.send('Got a POST request');
});
```

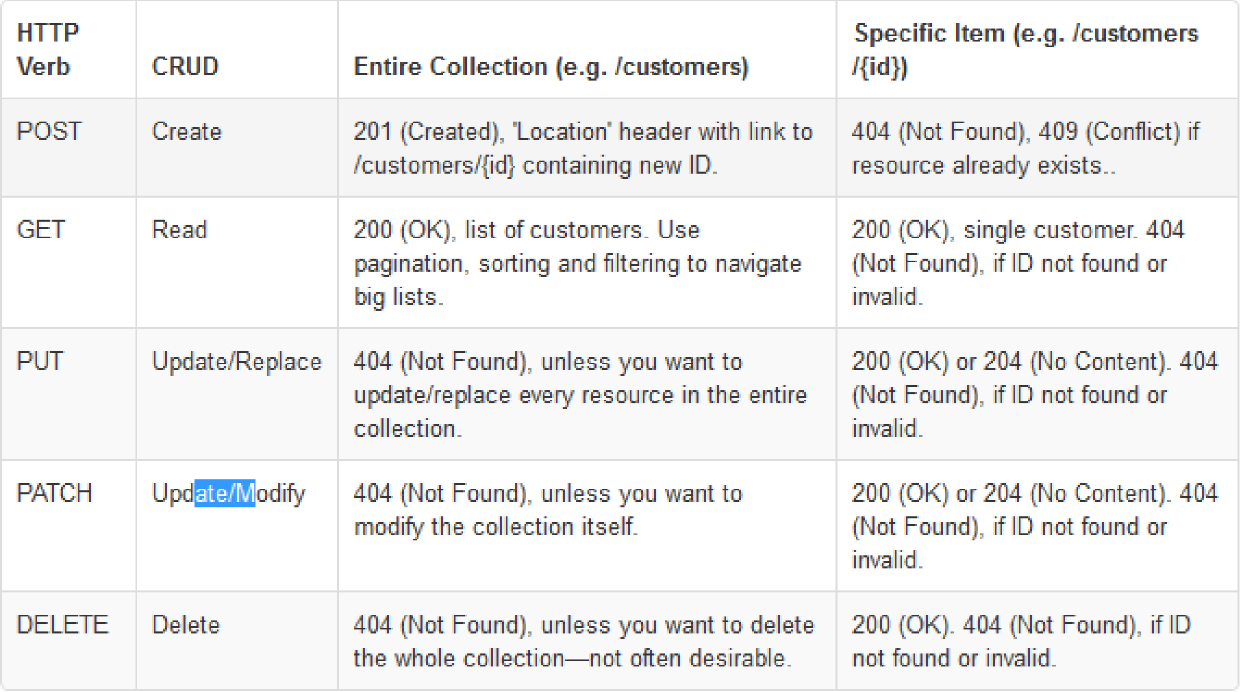


```
app.put('/user', function (req, res) {
  res.send('Got a PUT request at /user');
});
```



```
app.delete('/user', function (req, res) {
  res.send('Got a DELETE request at /user');
});
```

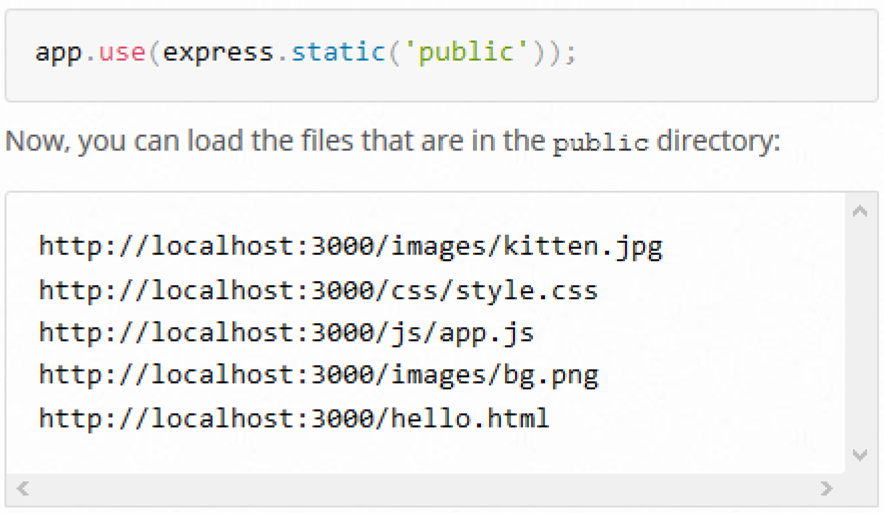
Lưu ý các dạng Request từ client.



HTTP Verb	CRUD	Entire Collection (e.g. /customers)	Specific Item (e.g. /customers/{id})
POST	Create	201 (Created), 'Location' header with link to /customers/{id} containing new ID.	404 (Not Found), 409 (Conflict) if resource already exists..
GET	Read	200 (OK), list of customers. Use pagination, sorting and filtering to navigate big lists.	200 (OK), single customer. 404 (Not Found), if ID not found or invalid.
PUT	Update/Replace	404 (Not Found), unless you want to update/replace every resource in the entire collection.	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid.
PATCH	Update/Modify	404 (Not Found), unless you want to modify the collection itself.	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid.
DELETE	Delete	404 (Not Found), unless you want to delete the whole collection—not often desirable.	200 (OK). 404 (Not Found), if ID not found or invalid.

Tham khảo: http://www.restapitutorial.com/lessons/httpmethods.html

3 – Cấu hình Static file

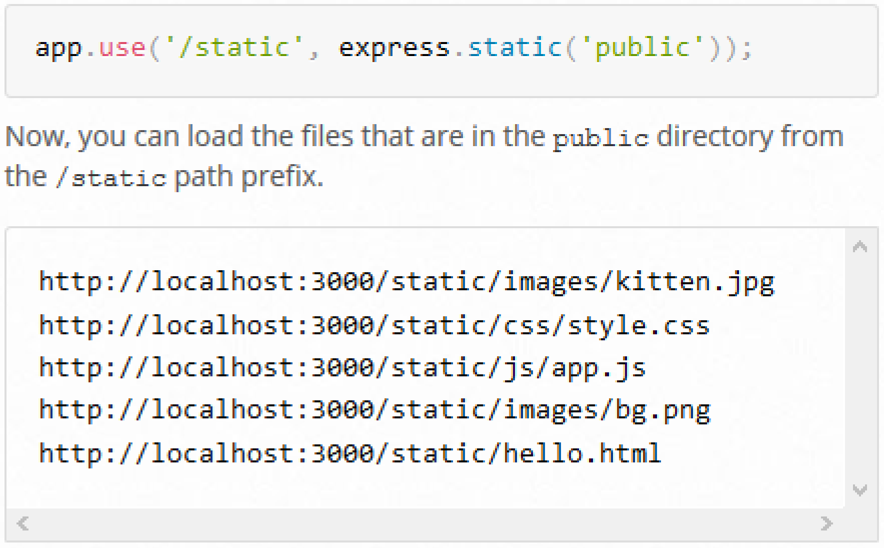


```
app.use(express.static('public'));
```

Now, you can load the files that are in the `public` directory:

```
http://localhost:3000/images/kitten.jpg
http://localhost:3000/css/style.css
http://localhost:3000/js/app.js
http://localhost:3000/images/bg.png
http://localhost:3000/hello.html
```

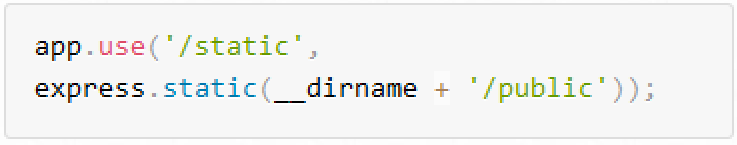
Lưu ý: Virtual static path



```
app.use('/static', express.static('public'));
```

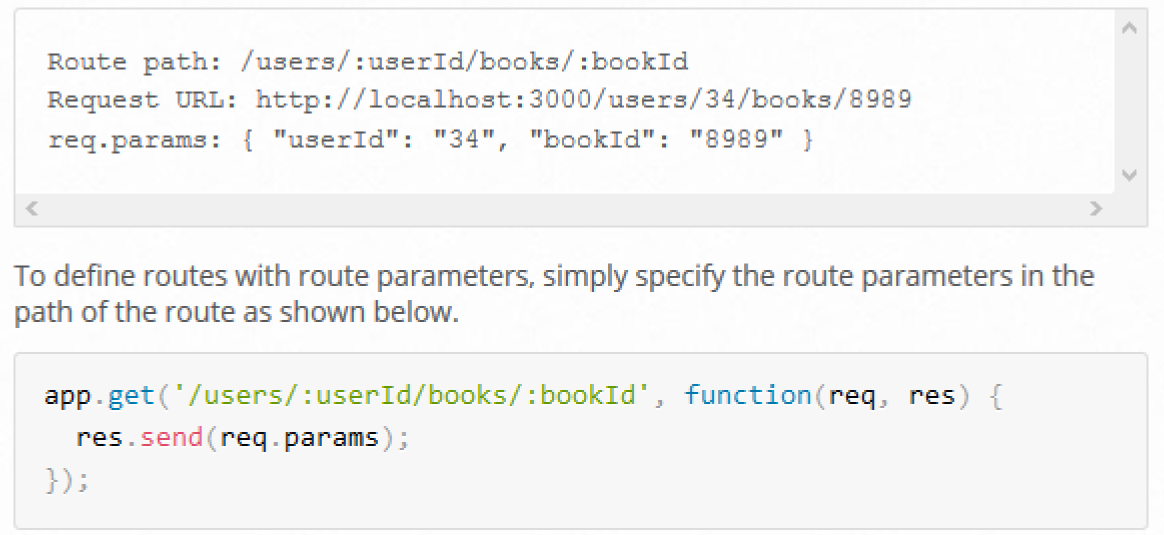
Now, you can load the files that are in the `public` directory from the `/static` path prefix.

```
http://localhost:3000/static/images/kitten.jpg
http://localhost:3000/static/css/style.css
http://localhost:3000/static/js/app.js
http://localhost:3000/static/images/bg.png
http://localhost:3000/static/hello.html
```



```
app.use('/static',
express.static(__dirname + '/public'));
```

4 – Truyền tham số với Route: Route Parameters

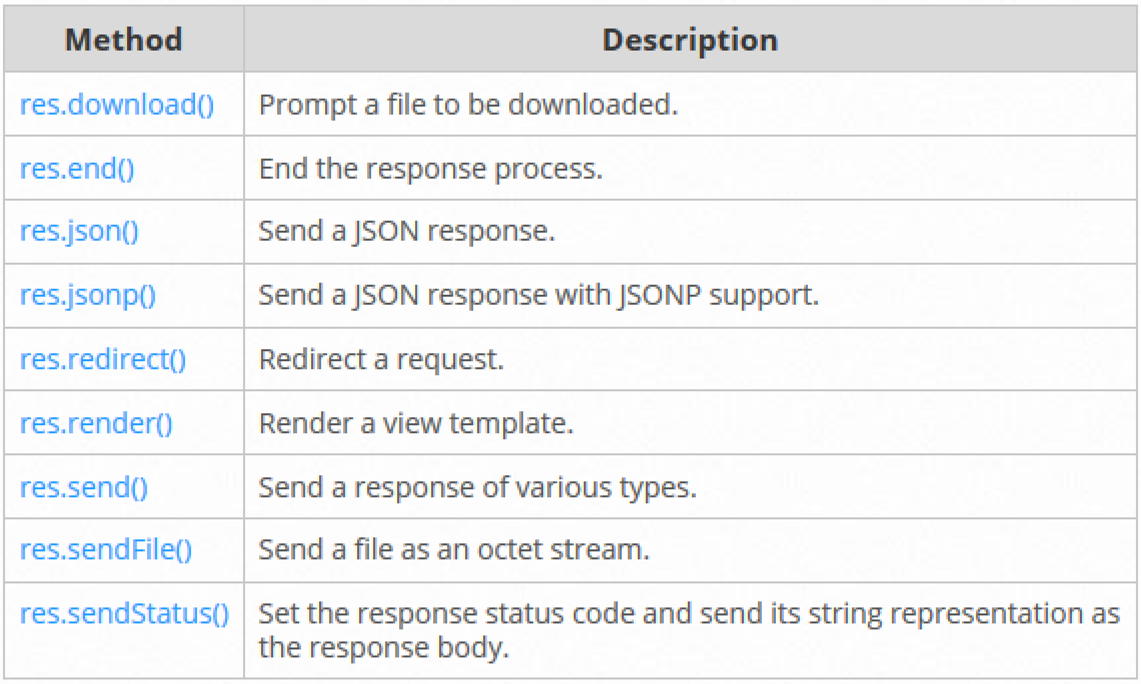


```
Route path: /users/:userId/books/:bookId
Request URL: http://localhost:3000/users/34/books/8989
req.params: { "userId": "34", "bookId": "8989" }
```

To define routes with route parameters, simply specify the route parameters in the path of the route as shown below.

```
app.get('/users/:userId/books/:bookId', function(req, res) {
  res.send(req.params);
});
```

5 - Respond Methods



Method	Description
res.download()	Prompt a file to be downloaded.
res.end()	End the response process.
res.json()	Send a JSON response.
res.jsonp()	Send a JSON response with JSONP support.
res.redirect()	Redirect a request.
res.render()	Render a view template.
res.send()	Send a response of various types.
res.sendFile()	Send a file as an octet stream.
res.sendStatus()	Set the response status code and send its string representation as the response body.